

Исследование производительности задачи поиска вширь в графе на сопроцессорах семейства Intel Xeon Phi

Семенов А.С., Головина Е.А., Фролов А.С.

5 марта 2014

Научно-технический семинар GraphHPC-2014

Задачи обработки графов, Intel Xeon Phi

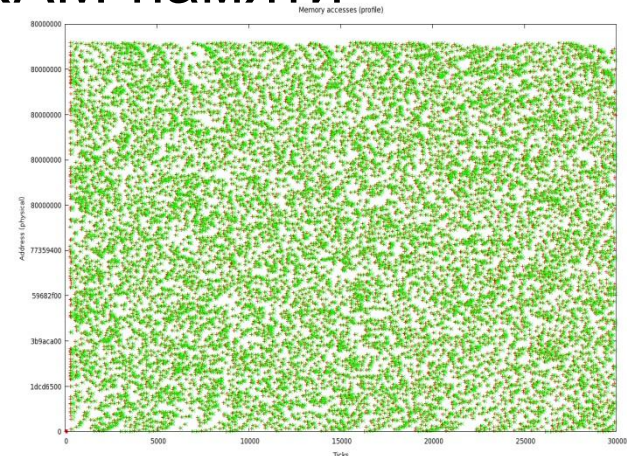
Intel Xeon Phi

- мультитредовый ускоритель
- длинные вектора



Задачи обработки больших графов

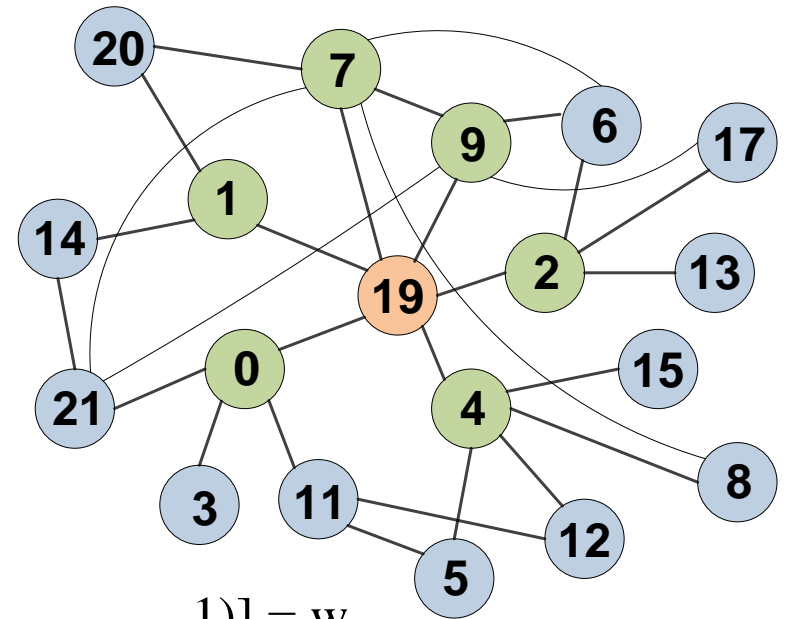
- неэффективное использование иерархии кэш-памяти
- большое количество промахов в TLB
- неэффективная работа DRAM-памяти



Поиск вширь в графе (BFS)

Подход Queue-based, алгоритм simple

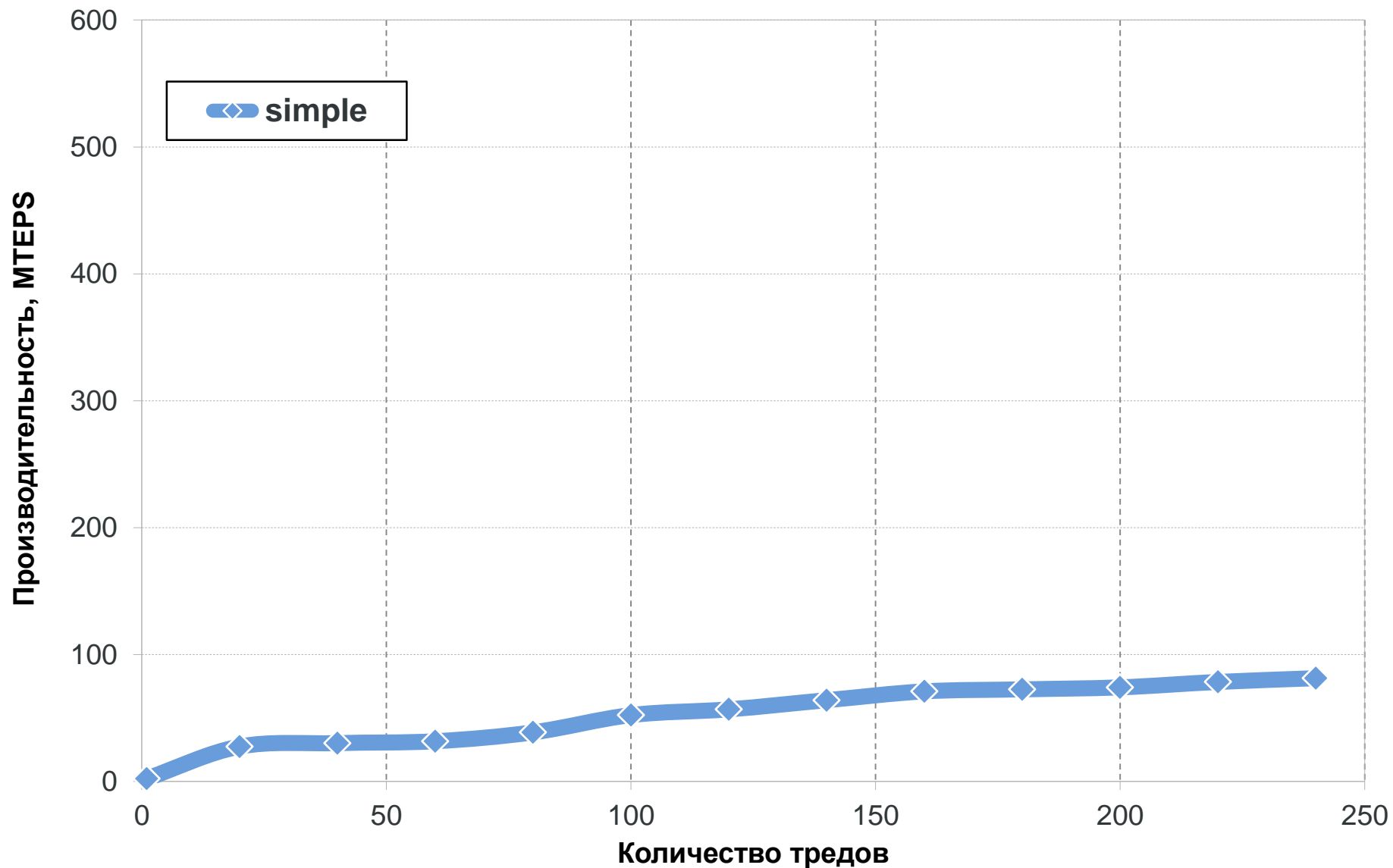
```
Q_counter = 1
Q[0] = root
marked[root] = 1
while Q_counter > 0
    Q_next_counter = 0
    #pragma omp parallel for
    for all vertex ∈ Q do
        for all w: (vertex, w) ∈ E do
            if marked[w] == 0 then
                Q_next[__sync_fetch_and_add(Q_next_counter, 1)] = w
                marked[w] = 1
            endif
        end for
    end for
    swap(Q, Q_next) // обмен Q и Q_next
end while
```



Характеристики используемых процессоров

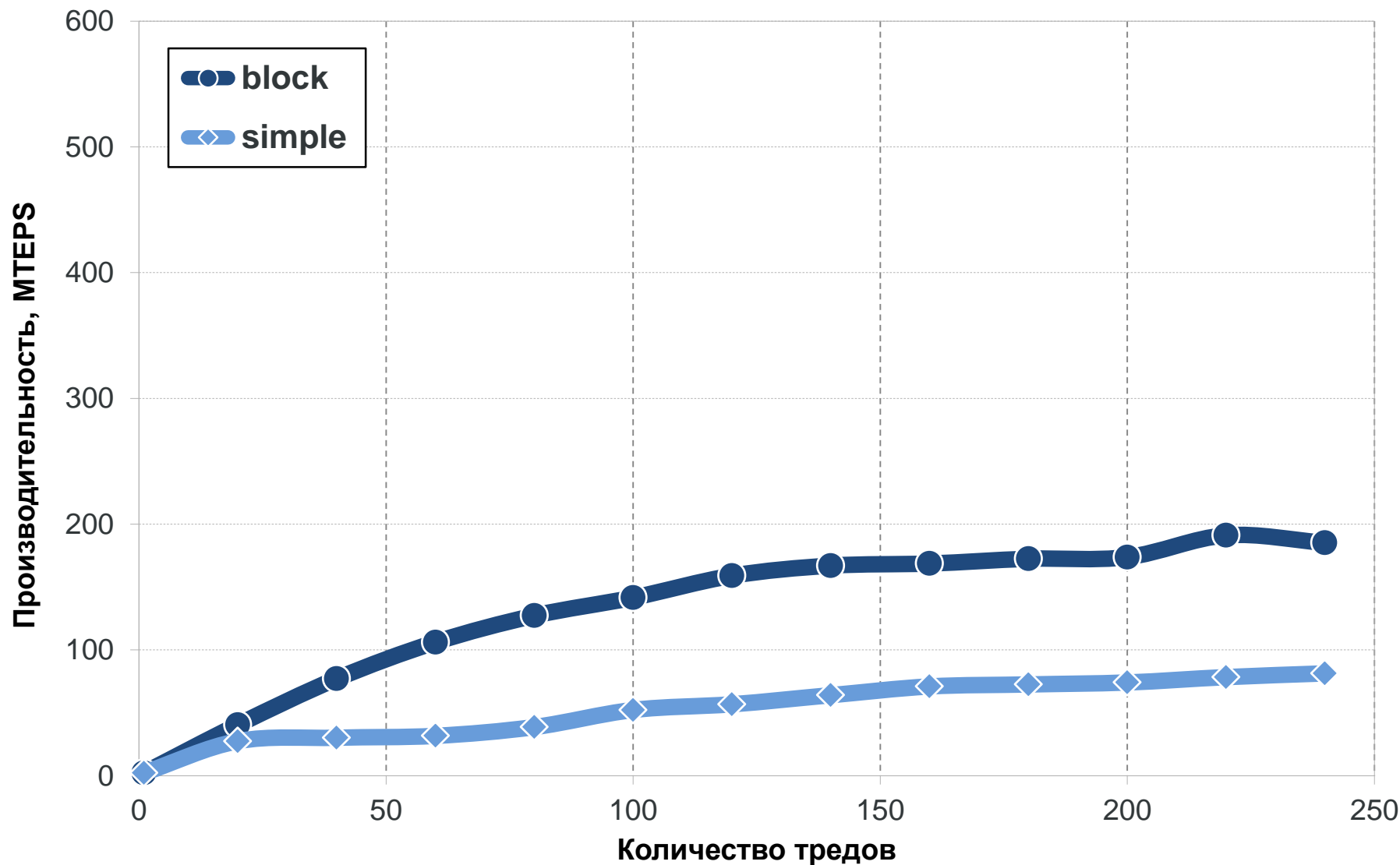
	Sandy Bridge-EP	Phi-5110P	Phi-7120P
Название модели	Xeon E5-2660	Xeon Phi 5110P	Xeon Phi 7120P
Частота	2.2 ГГц	1.05 ГГц	1.238
Количество сокетов	1	1	1
Количество ядер	8	60	61
Количество аппаратных тредов в ядре	2	4	4
Размер памяти кэшей данных (* - на одно ядро)	64 КБ* / 2 МБ* / 20 МБ	32 КБ* / 512 КБ*	32 КБ* / 512 КБ*
Объем установленной памяти	32 ГБ	8 ГБ	16
Тип используемой памяти	DDR3	GDDR5	GDDR5
Пиковая пропускная способность памяти	51 ГБ/с	352 ГБ/с	352 ГБ/с
Пропускная способность чтения из памяти, ГБ/с; Последовательный / случайный доступ	42 / 3.3	183 / 3.8	
Задержка обращения в память	~200 тактов	~300 тактов	~350 тактов

Производительность алгоритма simple в зависимости от числа используемых тредов на сопроцессоре Phi-5110P



Число вершин в графе: $N = 2^{27}$ (134 млн), средняя связность вершины: $k = 8$

Производительность алгоритмов simple и block в зависимости от числа используемых тредов на сопроцессоре Phi-5110P

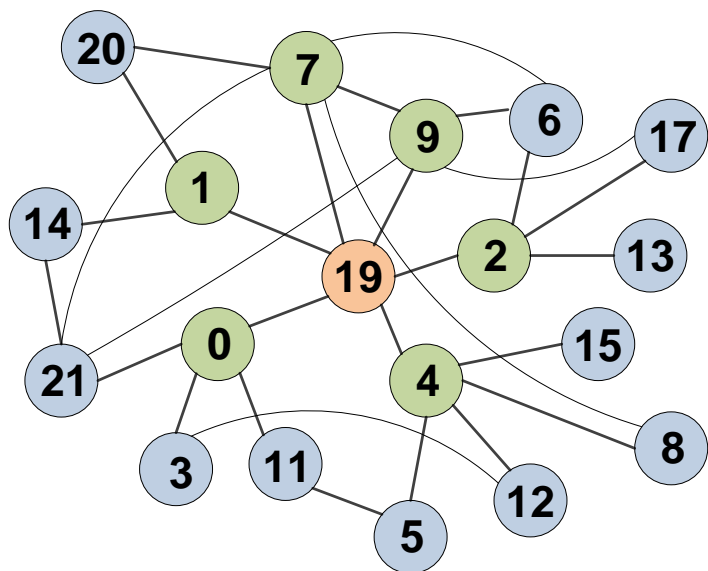


Число вершин в графе: $N = 2^{27}$ (134 млн), средняя связность вершины: $k = 8$

```
#pragma omp parallel for  
for all vertex  $\in Q$  do  
    for all w: (vertex, w)  $\in E$  do  
        if marked[w] == 0 then  
             $Q_{\text{next}}[\text{\_\_\text{sync\_fetch\_and\_add}(Q_{\text{next\_counter}}, 1)}] = w$   
            marked[w] = 1  
        endif  
    end for  
end for
```

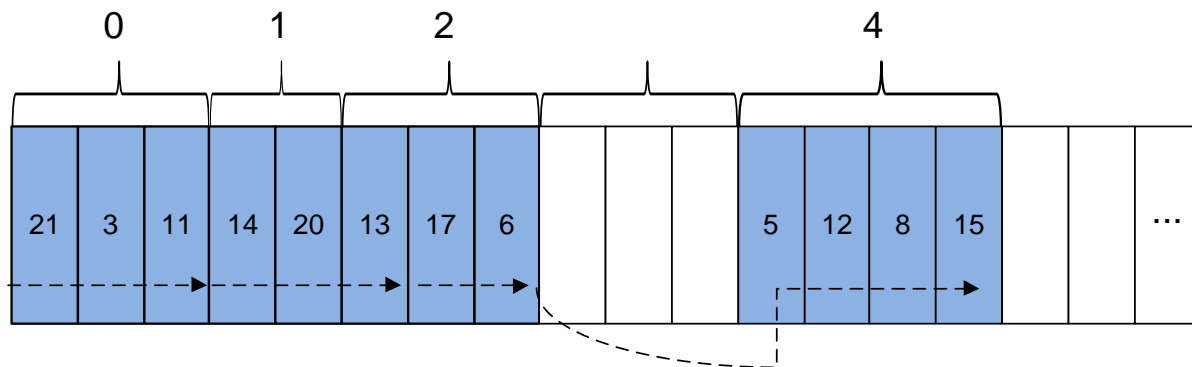


Подход Read-based, алгоритм read



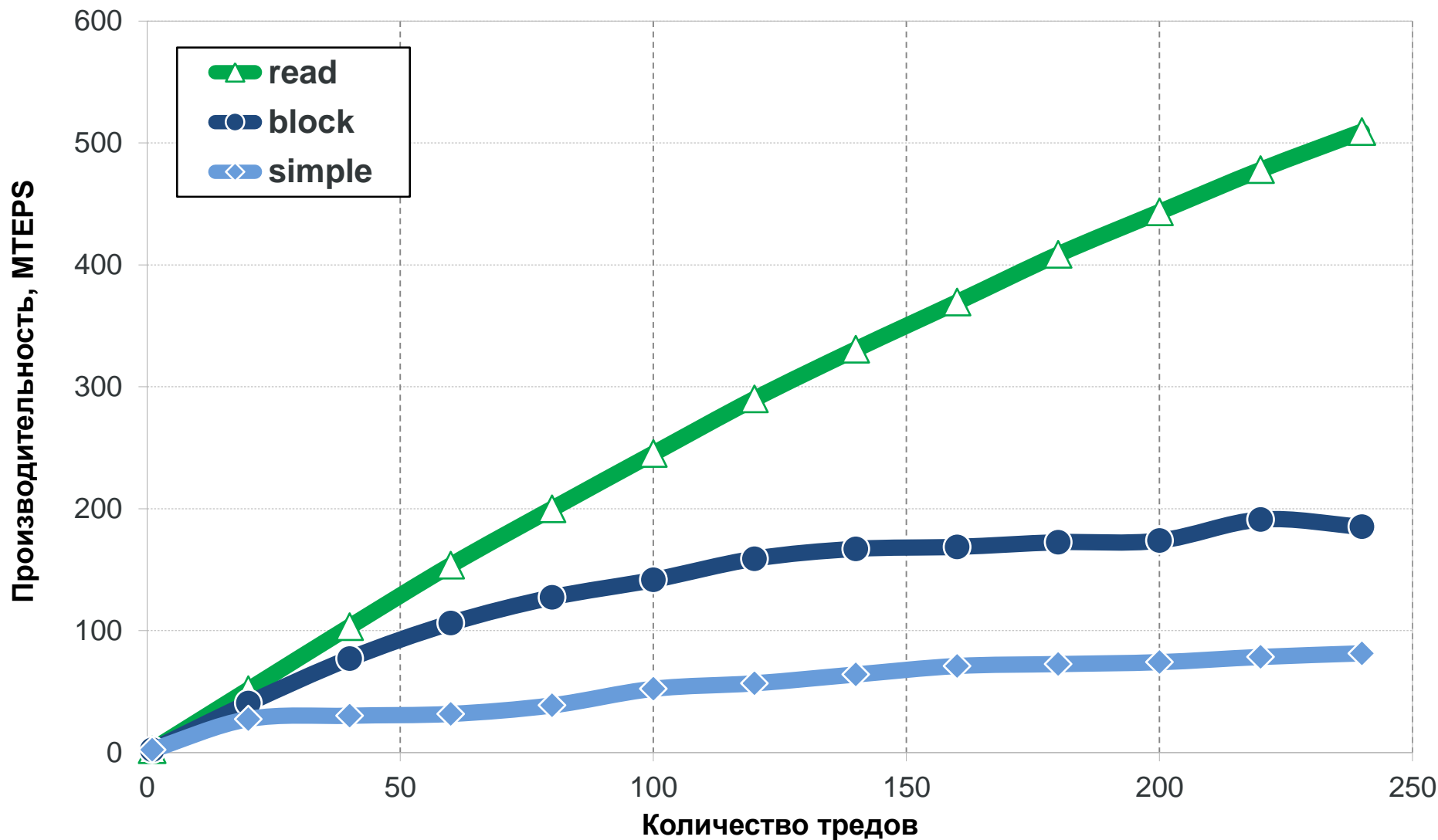
```
#pragma omp parallel for reduction (...)
for all vertex  $\in V$  do
  if levels[vertex]  $\neq$  numLevel then continue
  for all  $w$ : (vertex,  $w$ )  $\in E$  do
    if levels[w] == -1 then
      levels[w] = numLevel + 1
      nLevelVerts = nLevelVerts + 1
    end if
  end for
end for
```

0	1	2	3	4	массив levels				
1	1	1	INF	1	INF	INF	1	0	1



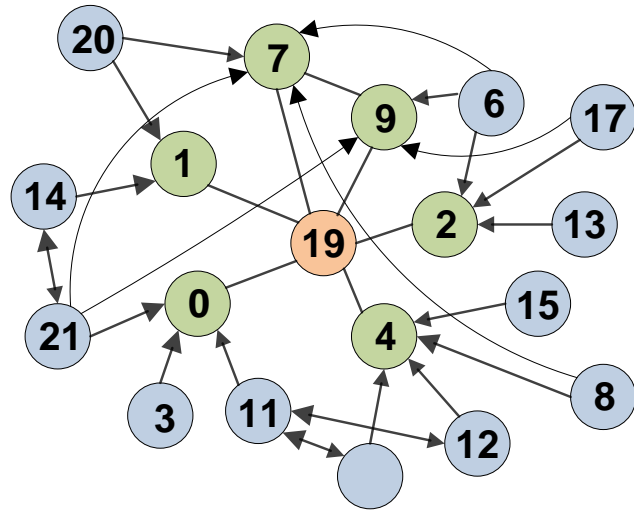
массив смежных вершин

Производительность алгоритмов simple, block и read в зависимости от числа используемых тредов на сопроцессоре Phi-5110P



Число вершин в графе: $N = 2^{27}$ (134 млн), средняя связность вершины: $k = 8$

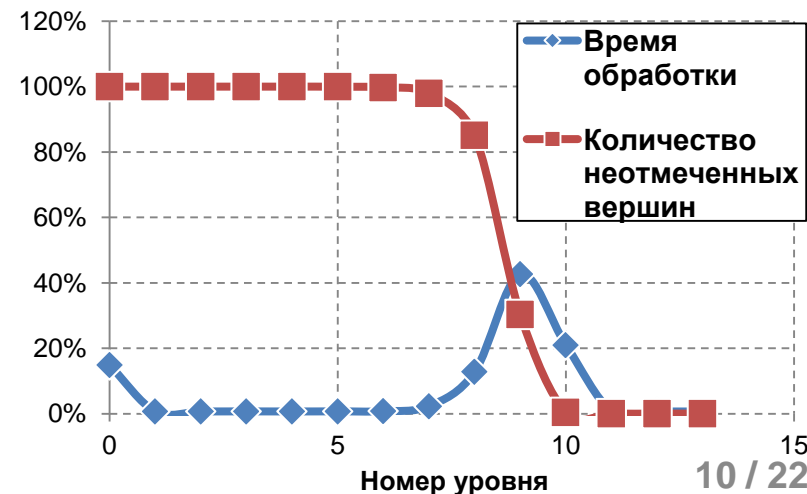
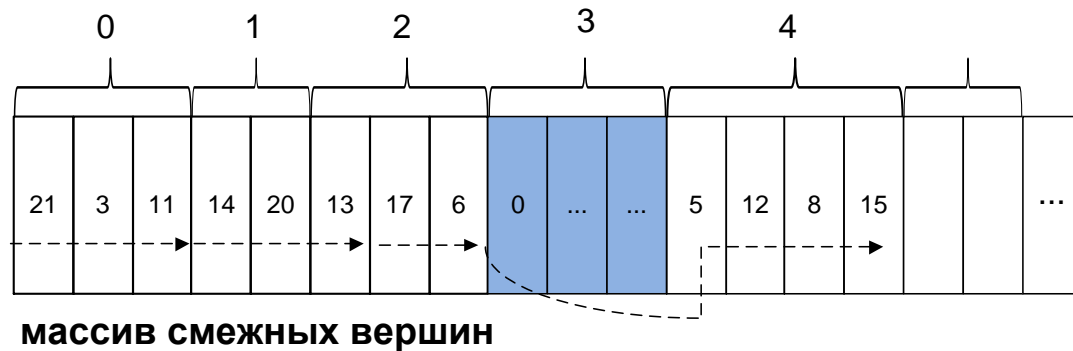
Алгоритм bottom-up-hybrid



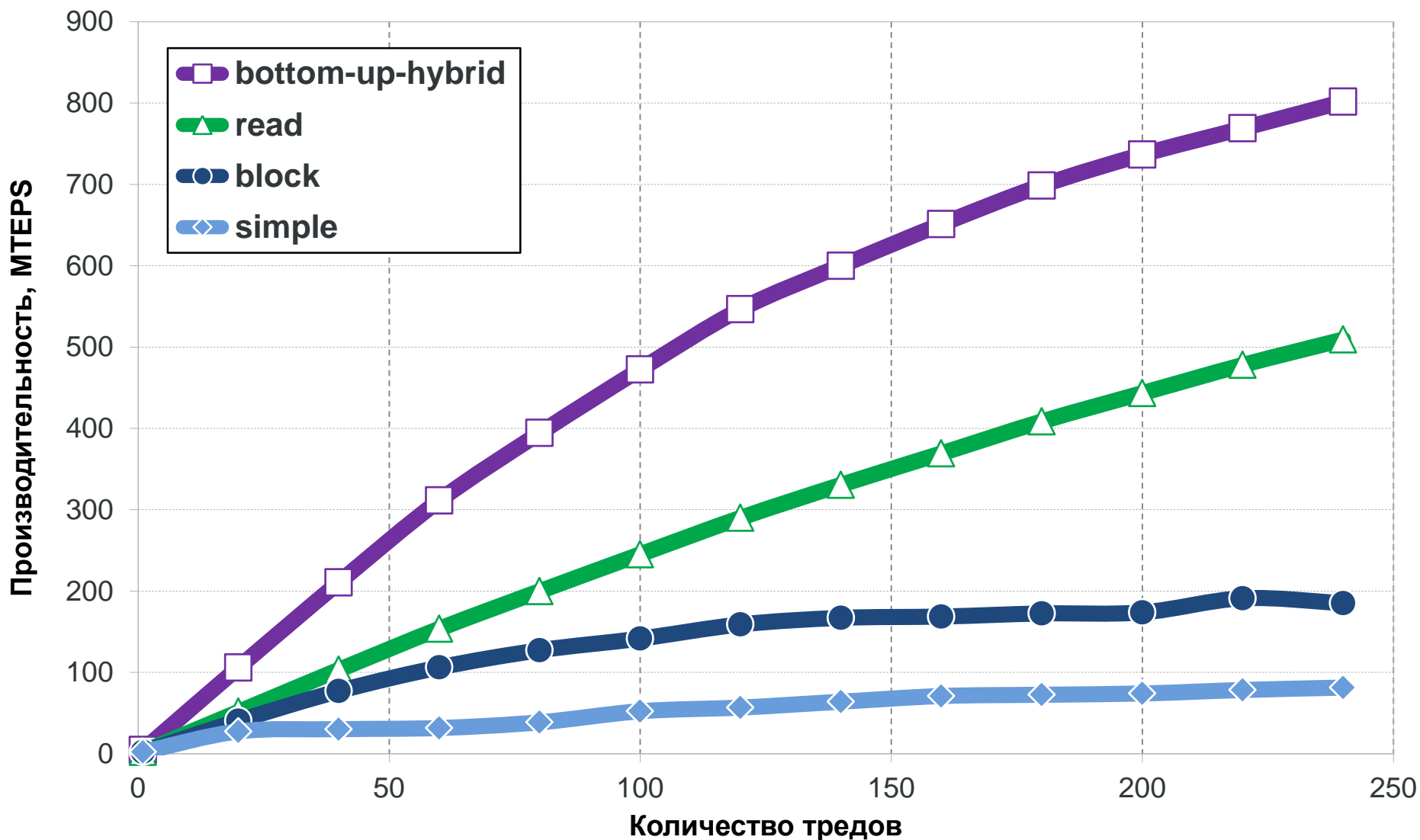
```

#pragma omp parallel for reduction (...)
for all vertex ∈ V do
    if levels[vertex] == -1 then
        for all w: (vertex, w) ∈ E do
            if levels[w] == numLevel then
                levels[vertex] = numLevel + 1
                nLevelVerts = nLevelVerts + 1
                break
            end if
        end for
    end if
end for
end if
end for
    
```

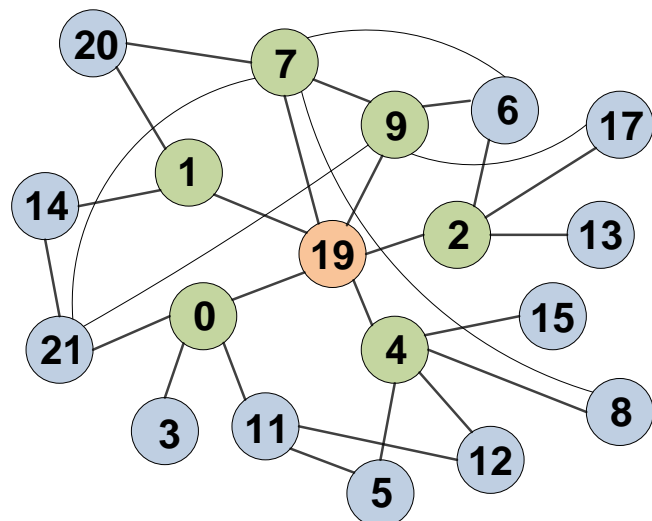
0	1	2	3	4	массив levels				
1	1	1	INF	1	INF	INF	1	0	1



Производительность алгоритмов simple, block, read и bottom-up-hybrid в зависимости от числа используемых тредов на сопроцессоре Phi-5110P

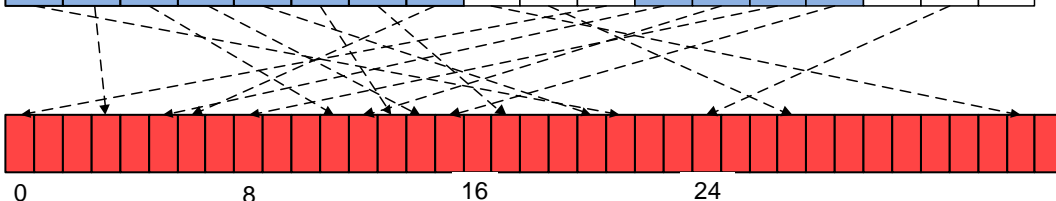


Недостатки алгоритмов read и bottom-up-hybrid



0	1	2	3	4	массив levels				
1	1	1	INF	1	INF	INF	1	0	1

массив смежных вершин									
0		1		2		3		4	
21	3	11	14	20	13	17	6	5	12
8	15	16	18	19	2	7	9	10	1

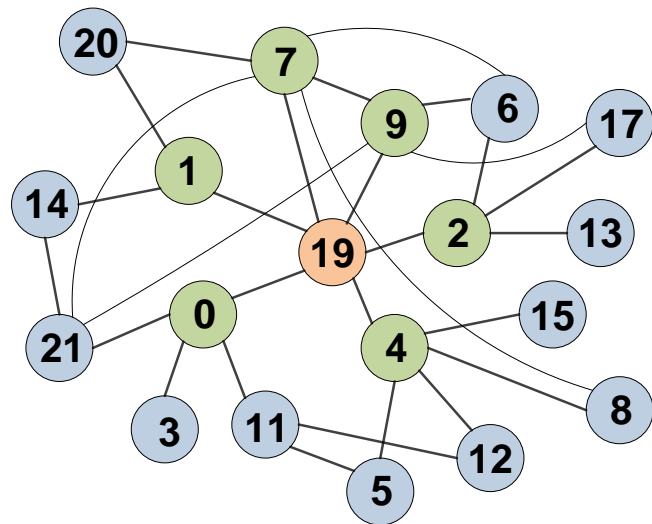


массив levels

```
#pragma omp parallel for reduction (...)
for all vertex ∈ V do
  if levels[vertex] ≠ numLevel then continue
  for all w: (vertex, w) ∈ E do
    if levels[w] == -1 then
      levels[w] = numLevel + 1
    ...
  end if
end for
end for
```

	SB	Phi-5110P
Частота, ГГц	2.2	1.05
Задержка обращения в память (такты)	~150	~300

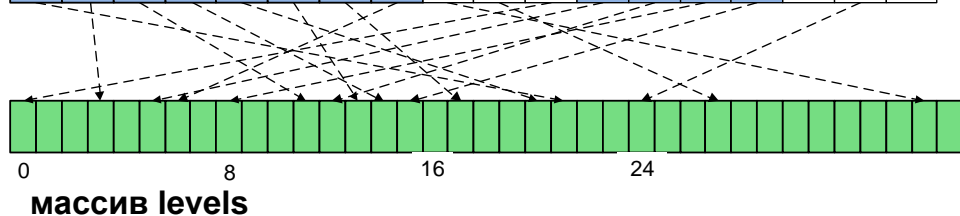
Решение: ручная развертка цикла + использование prefetch



0	1	2	3	4	массив levels					
1	1	1	INF	1	INF	INF	1	0	1	

массив смежных вершин

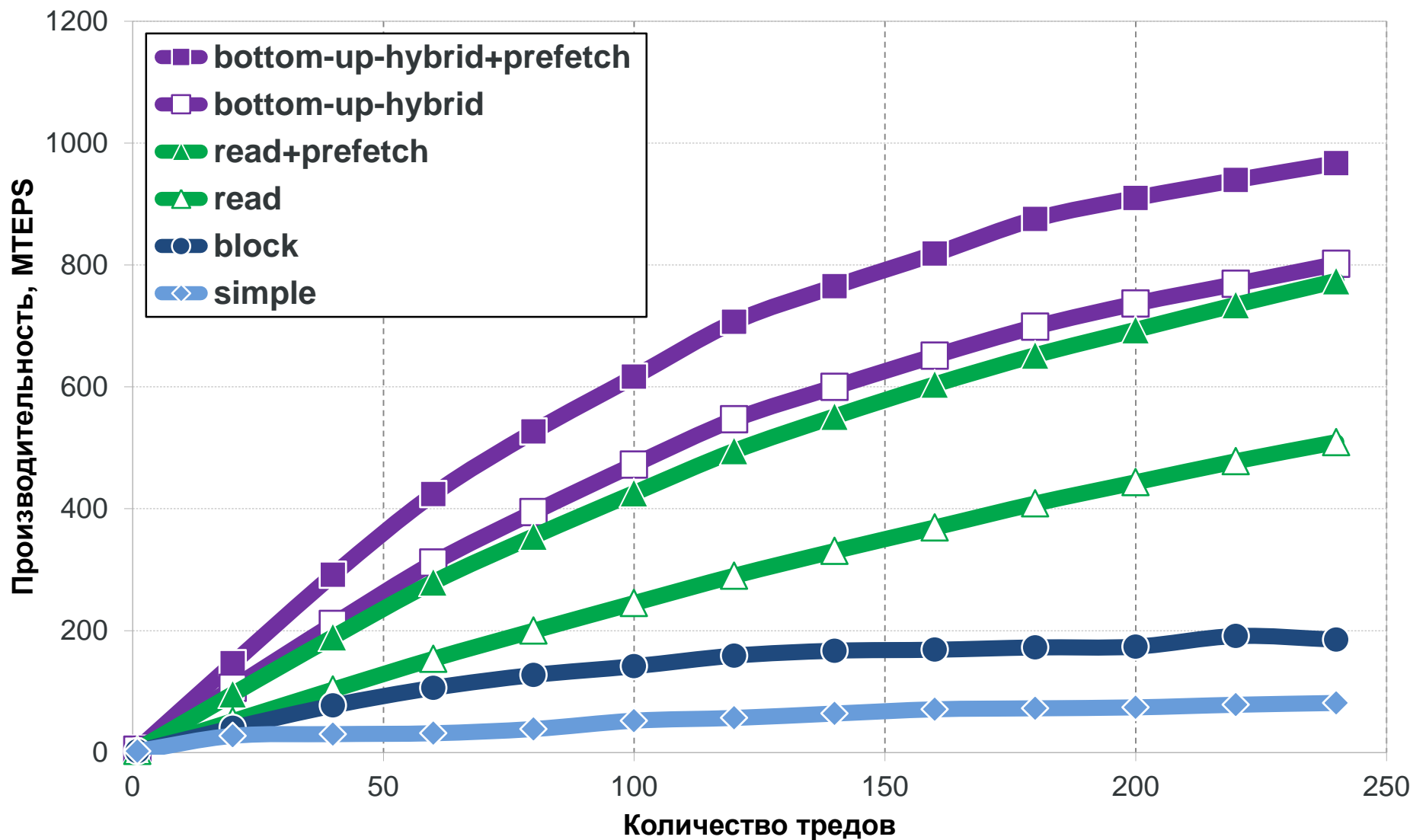
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	...
21	3	11	14	20	13	17	6															



```
#pragma omp parallel for reduction (...)
for all vertex ∈ V do
  if levels[vertex] ≠ numLevel then continue
  for all w: (vertex, w) ∈ E do
    prefetch(levels[w])
    ...
  if levels[w] == -1 then
    levels[w] = numLevel + 1
    ...
  end if
end for
end for
```

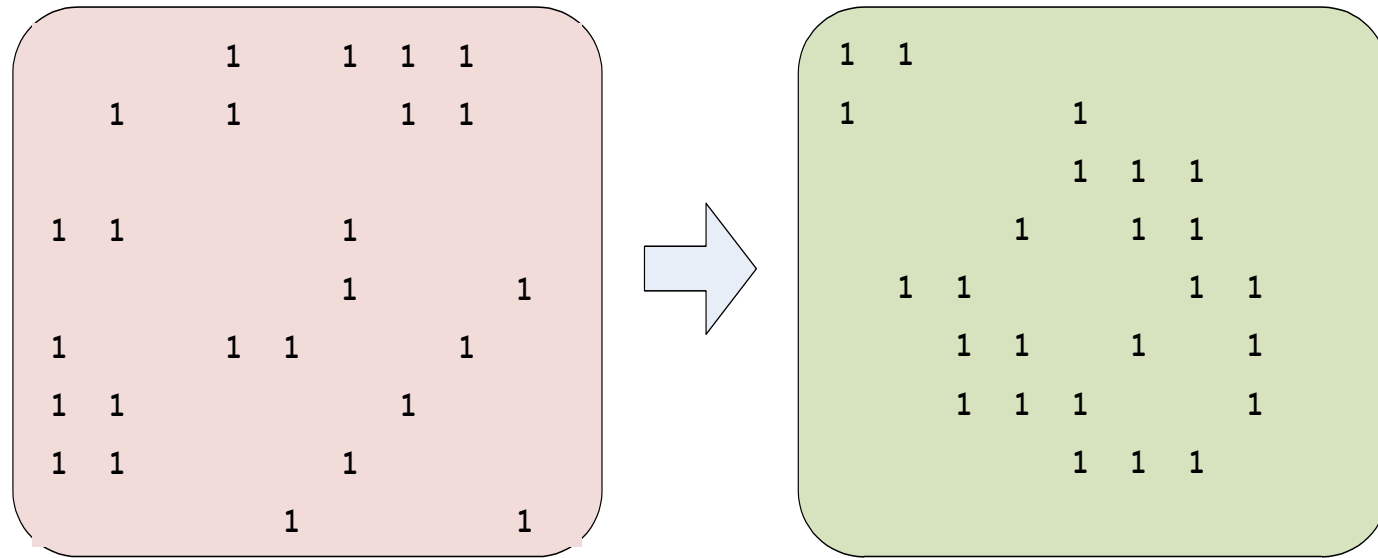
	SB	Phi-5110P
Пиковая пропускная способность памяти, ГБ/с	51	352
ПС чтения из памяти, ГБ/с; Последовательный / случайный доступ	42 / 3.3	183 / 3.8

Производительность алгоритмов simple, block, read и bottom-up-hybrid с префетчем в зависимости от числа используемых тредов на сопроцессоре Phi-5110P



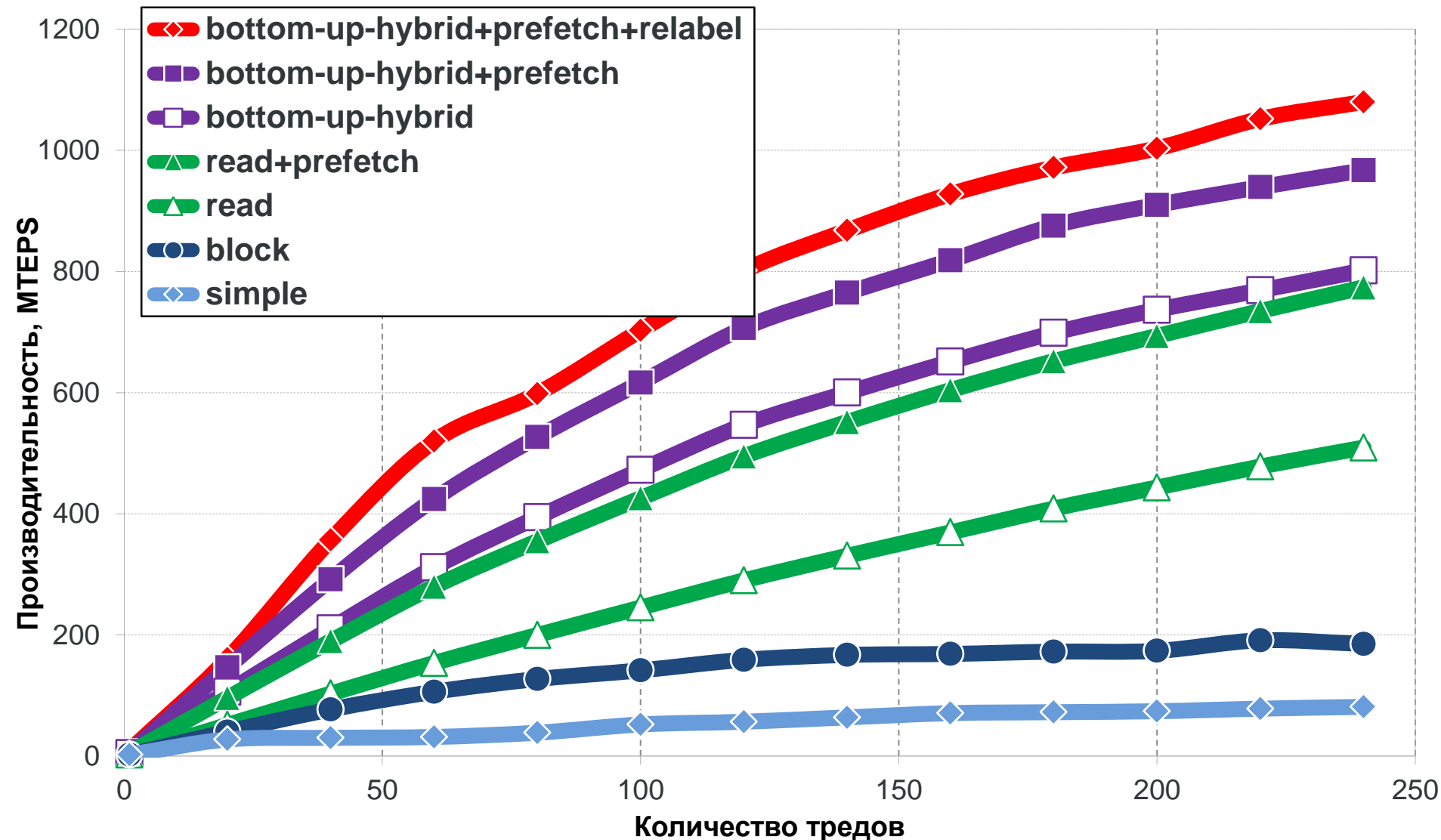
Число вершин в графе: $N = 2^{27}$ (134 млн), средняя связность вершины: $k = 8$

Улучшение локализации: перестановка вершин



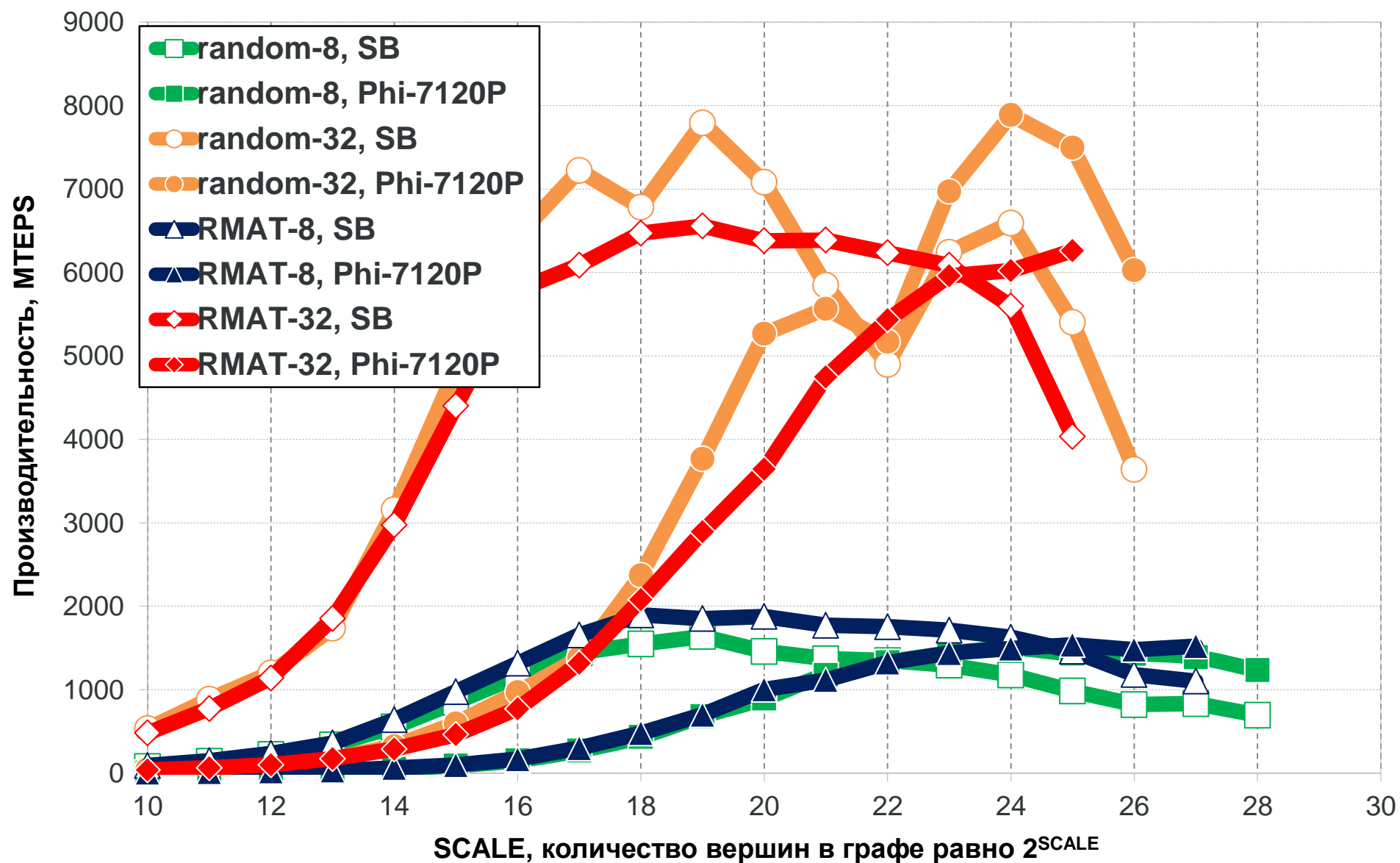
- Матрица смежности приводится к ленточному виду с уменьшением ширины ленты (алгоритм Reverse Cuthill-McKee) => уменьшается количество кэш-промахов
- Списки смежных вершин сортируются => уменьшается количество промахов в TLB
- Использование библиотеки libhugetlbfs больших страниц

Производительность различных алгоритмов, с префетчем и перестановками в зависимости от числа используемых тредов на сопроцессоре Phi-5110P

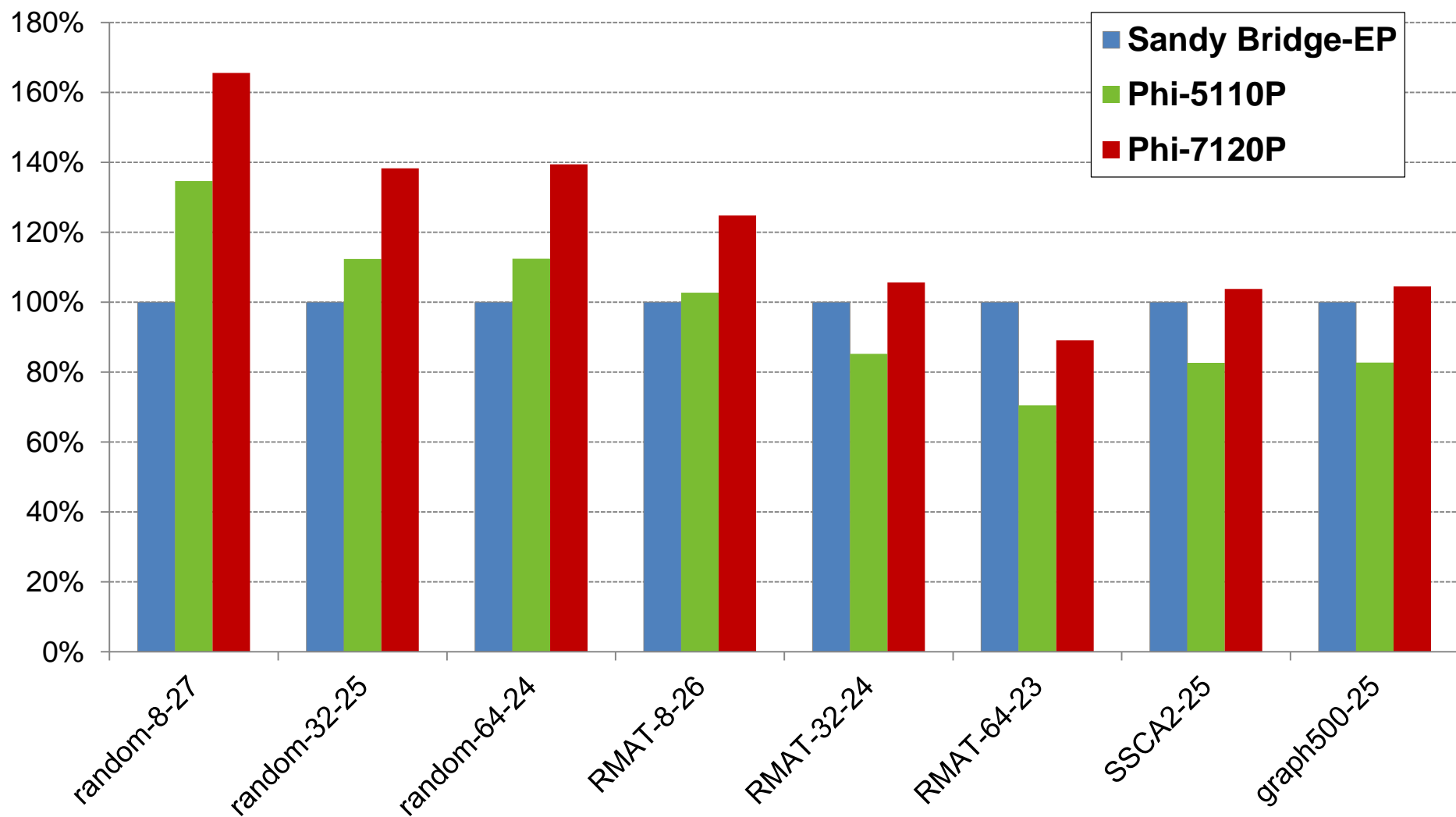


Число вершин в графе: $N = 2^{27}$ (134 млн), средняя связность вершины: $k = 8$

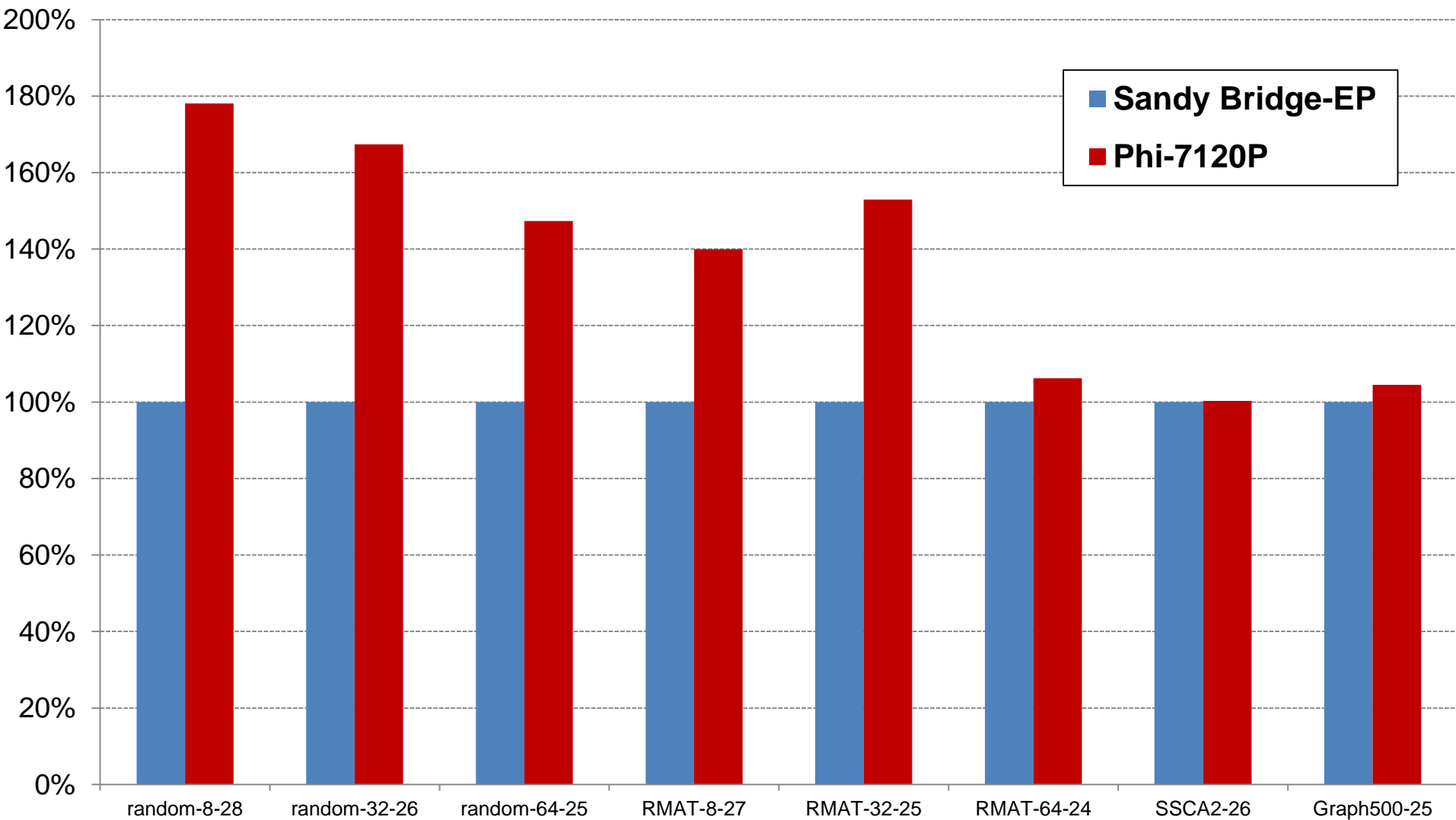
Сравнение производительности, достигаемой на Sandy Bridge-EP (SB) и Phi-7120P в зависимости от размера графа



Общее сравнение производительности, достигаемой на Sandy Bridge-EP (SB), Phi-5110P и Phi-7120P на разных графах максимального размера, помещающихся в 8 ГБ



Общее сравнение производительности, достигаемой на Sandy Bridge-EP (SB) и Phi-7120P на разных графах максимального размера, помещающихся в 16 ГБ



Сравнение с другими реализациями: список Graph500

- Ноябрь 2013:
 - 4366 MTEPS
 - граф 32 млн вершин (SCALE = 25)
 - 89 место
 - среди одноузловых систем 4-е место среди исследовательских групп
 - других результатов с Intel Xeon Phi нет

No	Rank	Machine	Installation Site	# Nodes	# Cores	SCALE	GTEPS
51	51	4xIntel Xeon E5-4650	Chuo University	1	64	27	31.65
58	58	Xeon E5-2650 v2, GeForce GTX TITAN	University of Tsukuba	1	8	25	17.21
70	70	TH-IVB-FEP/C - SMP	Changsha, China	1	24	26	9.74
89	89	MIC (Intel - Xeon Phi 7120P)	NICEVT, www.nicevt.ru	1	61	25	4.37

Заключение

- Наиболее эффективный алгоритм для Intel Xeon Phi:
 - максимальное использование пропускной способности памяти
 - последовательный и нерегулярный доступ
 - отсутствие атомарных операций
 - ручная развертка цикла и преднакачка в кэш
- Intel Xeon Phi 5110P vs Intel Sandy Bridge-EP:
максимальный выигрыш – **34%**, максимальный проигрыш **29%**
- Intel Xeon Phi 7120P vs Intel Sandy Bridge-EP:
максимальный выигрыш – **78%**, в среднем – выигрыш **37%**
- Приятное впечатление от Intel Xeon Phi, побольше бы памяти...
- Дальнейшая оптимизация – инструкция `vpgatherdd`?



Спасибо за внимание!

Вопросы?

alxdr.semenov@gmail.com